Computing Least Squares Objective Function Sensitivities for ODEs and DDEs

Jonathan Calver
(Supervised by Wayne Enright)

Department of Computer Science
University of Toronto

May 7, 2015

Outline

Motivation

- Mathematical models contain parameters
- Estimating these parameters often involves a least squares (LSQ) minimization, where we want to minimize the sum of squared errors between the **model predictions** and **observations**
- If the model is non-linear, this optimization requires gradient information (**sensitivities**)

Least Squares Objective Function

$$J(\mathbf{p}) = \sum_{i=1}^{n_o} \sum_{j=1}^{n_y} \frac{\left(\tilde{\mathbf{y}}_j(t_i) - \mathbf{y}_j(t_i, \mathbf{p})\right)^2}{2}$$

- $\tilde{\mathbf{y}}(t_i) =$ observation at time t_i
- $\mathbf{y}(t_i, \mathbf{p}) =$ model prediction at time t_i
- $\mathbf{y}_j = j^{th}$ state variable
- $\mathbf{p} =$ vector of unknown parameters
- $n_y =$ number of state variables
- $n_p =$ number of parameters
- $n_o =$ number of observations

Least Squares Objective Function

$$J(\mathbf{p}) = \sum_{i=1}^{n_o} \sum_{j=1}^{n_y} \frac{\left(\tilde{\mathbf{y}}_j(t_i) - \mathbf{y}_j(t_i, \mathbf{p})\right)^2}{2}$$

To optimize, we require the sensitivities, $\frac{\partial J}{\partial \mathbf{p}}$. The rest of this talk is about numerically approximating $\frac{\partial J}{\partial \mathbf{p}}$, for the cases where $\mathbf{y}(t_i, \mathbf{p})$ satisfies:

- a system of Ordinary Differential Equations (ODEs)
- a system of Delay Differential Equations (DDEs)

Approximating Sensitivities

using Finite Differences (FD)

- Standard method for approximating the gradient of a function

$$J_p \approx \frac{J(p+\epsilon) - J(p)}{\epsilon} + \mathcal{O}(\epsilon)$$

- Requires J to be computed an additional time for each parameter
- Limited accuracy available, due to numerical issues

Definitions

ODEs

We consider the initial value problem (IVP),

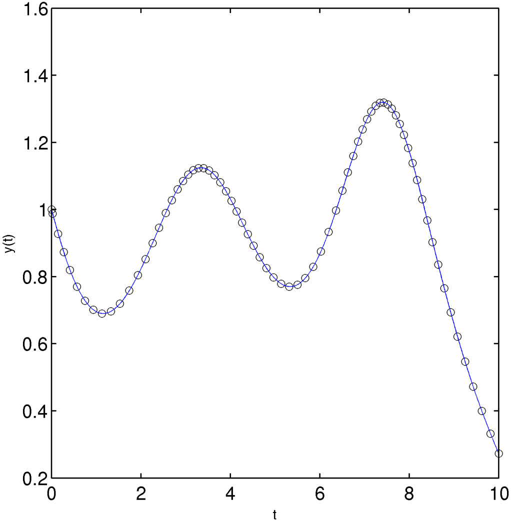
$$\begin{aligned}
&\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t), \mathbf{p}) \\
&\mathbf{y}(0) = \mathbf{y}_0 \\
&t \in (0, T)
\end{aligned}$$

Numerical Solution of ODE IVPs

- Starting from $t = 0$, advance the solution through time, such that $\mathbf{y}(t)$ approximately satisfies the ODE, up to a user specified error tolerance.
- Many numerical schemes exist for doing this
- We use a high order, explicit continuous Runge-Kutta (CRK) solver
- CRK provides a piece-wise continuous polynomial approximation to $\mathbf{y}(t)$ over the interval $[0, T]$.

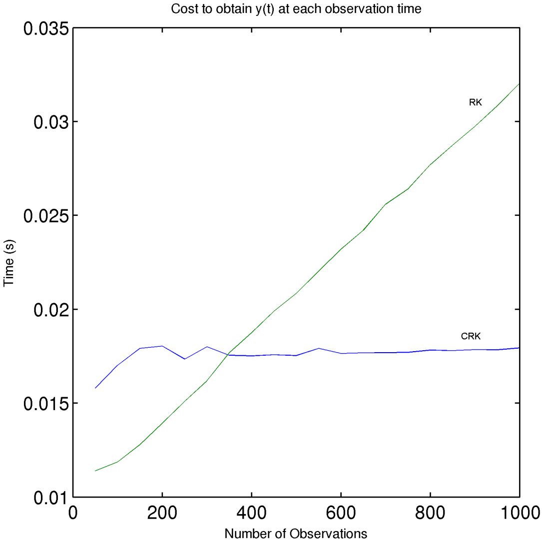
Example

Solution of an ODE IVP



Cost of Obtaining $\mathbf{y}(t)$

RK vs CRK



Variational Method

for ODEs

$$
\begin{aligned}
\frac{d}{dt}\mathbf{y}_{\mathbf{p}}(t) &= \frac{\partial}{\partial \mathbf{p}}\frac{d\mathbf{y}}{dt}(t) \\
&= \frac{\partial}{\partial \mathbf{p}}\mathbf{f}(t, \mathbf{y}(t, \mathbf{p}), \mathbf{p}) \\
&= \mathbf{f}_{\mathbf{y}}(t)\mathbf{y}_{\mathbf{p}}(t) + \mathbf{f}_{\mathbf{p}}(t)
\end{aligned}
$$

Initial conditions: $\mathbf{y}_{\mathbf{p}}(0)$, whose (k, j) entry is,

$$
\frac{\partial \mathbf{y}_j}{\partial \mathbf{p}_k}(0) = \begin{cases} 1, & \text{if } \mathbf{p}_k \text{ is the initial condition for } \mathbf{y}_j \\ 0, & \text{otherwise} \end{cases} .
$$

Variational Method
for ODEs

Solving the Variational Equations, we obtain $\mathbf{y_p}(t)$. Then we compute $J_{\mathbf{p}}$:

$$J_{\mathbf{p}} = \sum_{i=1}^{n_o} (\tilde{\mathbf{y}}_j(t_i) - \mathbf{y}_j(t_i, \mathbf{p}))^T \mathbf{y_p}(t_i, \mathbf{p})$$

- The system of variational equations is of size $n_y n_p$
- Directly approximates the **model sensitivities**, $\mathbf{y_p}(t)$

Least Squares Objective Function

for adjoint method

$$J(\mathbf{p}) = \int_0^T g(s, \mathbf{y}(s, \mathbf{p}))ds$$

$$g(s, \mathbf{y}(s, \mathbf{p})) = \sum_{i=1}^{n_o} \sum_{j=1}^{n_y} \frac{(\tilde{\mathbf{y}}_j(t_i) - \mathbf{y}_j(t_i, \mathbf{p}))^2}{2} \delta(t_i - s)$$

- $\delta(t_i - s)$ is the Dirac-delta function

Adjoint Method

for ODEs

The adjoint vector, $\boldsymbol{\lambda}^T(t)$, satisfies the IVP,

$$\dot{\boldsymbol{\lambda}}^T(t) = g_{\mathbf{y}}(t) - \boldsymbol{\lambda}^T(t)\mathbf{f}_{\mathbf{y}}(t)\,;\boldsymbol{\lambda}^T(T) = \mathbf{0}$$

$$J_{\mathbf{p}} = -\int_0^T \boldsymbol{\lambda}^T(t)\mathbf{f}_{\mathbf{p}}(t)\,dt - \boldsymbol{\lambda}^T(0)\mathbf{y}_{\mathbf{p}}(0)$$

See [1] for full derivation

Adjoint Method

for ODEs

The adjoint vector, $\boldsymbol{\lambda}^T(t)$, satisfies the IVP,

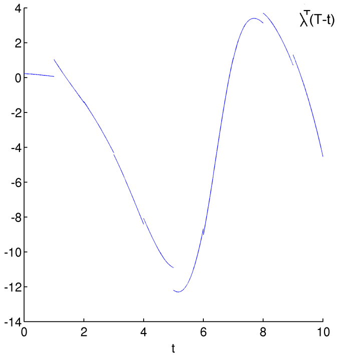
$$\dot{\boldsymbol{\lambda}}^T(t) = g_{\mathbf{y}}(t) - \boldsymbol{\lambda}^T(t)\mathbf{f}_{\mathbf{y}}(t)\,;\boldsymbol{\lambda}^T(T) = \mathbf{0}$$

$$J_{\mathbf{p}} = -\int_0^T \boldsymbol{\lambda}^T(t)\mathbf{f}_{\mathbf{p}}(t)\,dt - \boldsymbol{\lambda}^T(0)\mathbf{y}_{\mathbf{p}}(0)$$

We handle each $\delta(t_i - t)$, in $g_{\mathbf{y}}(t)$, by forcing the solver to hit $t = t_i$ and applying the jump:

$$\boldsymbol{\lambda}^T(t_i)^- = \boldsymbol{\lambda}^T(t_i)^+ + (\tilde{\mathbf{y}}(t_i) - \mathbf{y}(t_i, \mathbf{p}))$$

Solution of Adjoint System

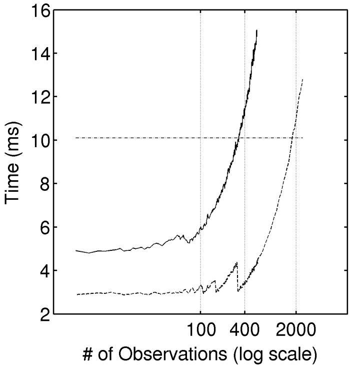


Method Comparison

	Finite Differences	Adjoint	Variational
simulations	$n_p + 1$	2	1
system size	n_y	n_y, n_p	$n_y + n_y n_p$

Runtime vs Number of Observations

for ODEs



Definitions

DDEs

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t), \mathbf{y}(t-\tau_1), \ldots, \mathbf{y}(t-\tau_{n_l}), \mathbf{p})$$
$$t \in (0, T)$$
$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{p})\ ;\ t < 0$$

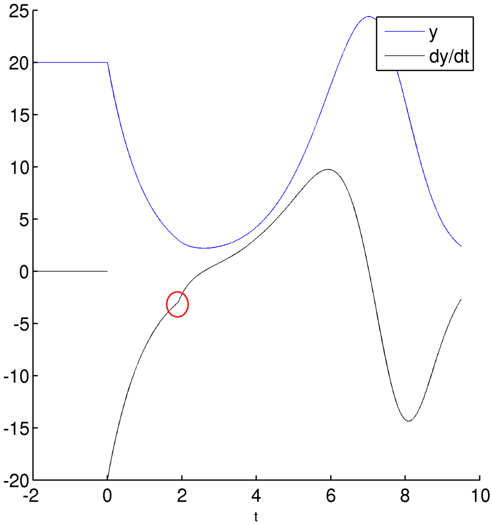
h is the history function

Numerical Solution of DDEs

- We use the DDEM package developed by Hossein Zivari Piran [3]
- To evaluate the lagged values of $\mathbf{y}$, we require a continuous approximation to be maintained
- Between discontinuity points, problem reduces to solving an ODE

In the following, we define $\boldsymbol{\nu} = \mathbf{y}(t - \tau)$.

Example
DDE



Variational Method

for DDEs

$$
\begin{aligned}
\frac{d}{dt}\mathbf{y}_{\mathbf{p}}(t) &= \frac{\partial}{\partial \mathbf{p}}\frac{d\mathbf{y}}{dt}(t) \\
&= \frac{\partial}{\partial \mathbf{p}}\mathbf{f}(t, \mathbf{y}(t,\mathbf{p}), \mathbf{y}(t-\tau,\mathbf{p}), \mathbf{p}) \\
&= \mathbf{f}_{\mathbf{y}}(t)\mathbf{y}_{\mathbf{p}}(t) + \mathbf{f}_{\mathbf{p}}(t) + \mathbf{f}_{\nu}(t)\mathbf{y}_{\mathbf{p}}(t-\tau) - \mathbf{y}'(t-\tau)\tau_{\mathbf{p}}
\end{aligned}
$$

Adjoint Method

for DDEs

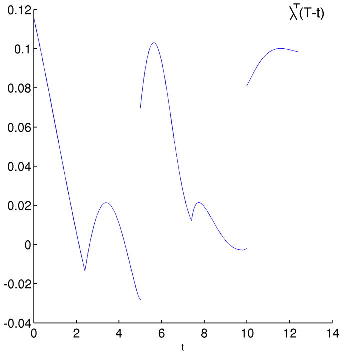
$$\dot{\boldsymbol{\lambda}}^T(t) = g_{\mathbf{y}}(t) - \boldsymbol{\lambda}^T(t)\mathbf{f}_{\mathbf{y}}(t) - \boldsymbol{\lambda}^T(t+\tau)\mathbf{f}_{\boldsymbol{\nu}}(t+\tau)\ ; \boldsymbol{\lambda}^T(t) = \mathbf{0}, \text{for } t \geq T.$$

$$\begin{aligned} J_{\mathbf{p}} &= -\int_0^T \boldsymbol{\lambda}^T(t)\Big(\mathbf{f}_{\boldsymbol{\nu}}(t)\mathbf{y}'(t-\tau)\alpha_{\mathbf{p}}(t) + \mathbf{f}_{\mathbf{p}}(t)\Big)\, dt \\ &\quad -\boldsymbol{\lambda}^T(0)\mathbf{y}_{\mathbf{p}}(0) - \int_{-\tau}^{0} \boldsymbol{\lambda}^T(t+\tau)\mathbf{f}_{\boldsymbol{\nu}}(t+\tau)\mathbf{h}_{\mathbf{p}}(t)\, dt \end{aligned}$$

See derivation in [4] for a more general class of DDEs

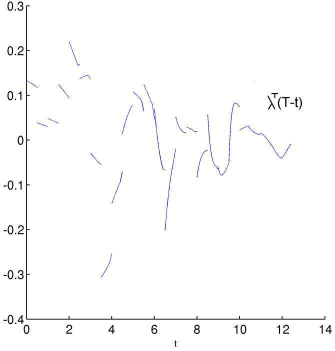
Solution of Adjoint System

3 observations



Solution of Adjoint System

20 observations



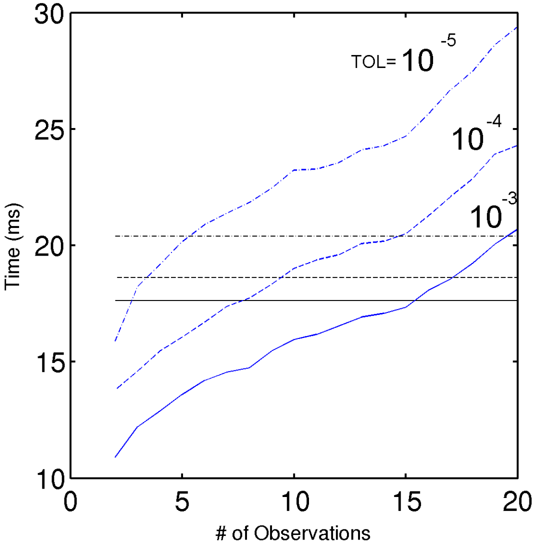
Tracking Discontinuities

Adjoint Steps	Time	Max Rel Error	Tol	Max Order
282	0.0369	5.39e-05	1e-05	0
182	0.0223	5.39e-05	1e-05	1
163	0.0186	5.39e-05	1e-05	**2**
175	0.019	5.39e-05	1e-05	3
200	0.0206	5.39e-05	1e-05	4
224	0.0223	5.39e-05	1e-05	5

To efficiently solve the adjoint system, we should track up to second order discontinuities

Runtime vs Number of Observations

for DDEs



Ways to Improve Performance of Adjoint Approach

for DDEs

- Use a lower order CRK solver for the adjoint system
- Remove the discontinuities in $\boldsymbol{\lambda}(t)$ by approximating the least squares objective function

Recall: Least Squares Objective Function

for adjoint method

$$J(\mathbf{p}) = \int_0^T g(s, \mathbf{y}(s, \mathbf{p}))ds$$

$$g(s, \mathbf{y}(s, \mathbf{p})) = \sum_{i=1}^{n_o} \sum_{j=1}^{n_y} \frac{(\tilde{\mathbf{y}}_j(t_i) - \mathbf{y}_j(t_i, \mathbf{p}))^2}{2} \delta(t_i - s)$$

- Can approximate $\delta(t_i - s)$

$$\delta(t_i - t) = \lim_{\sigma \to 0} \frac{1}{\sigma\sqrt{\pi}} e^{-(\frac{t - t_i}{\sigma})^2}$$

Approximation by Continuous Objective

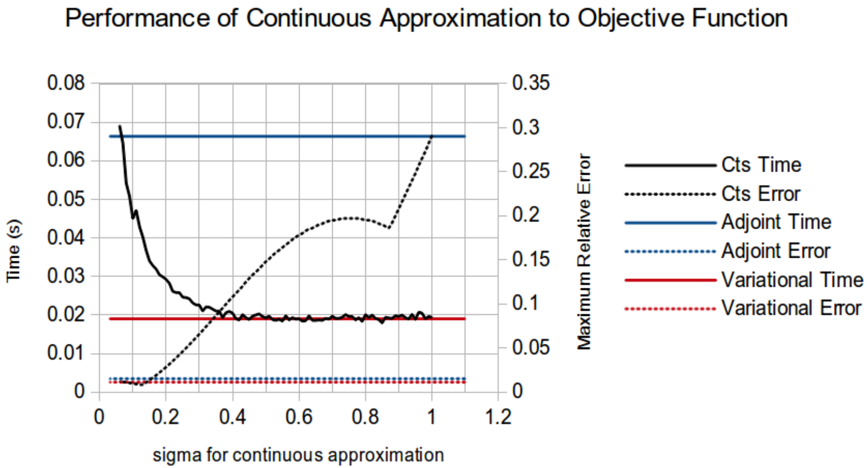
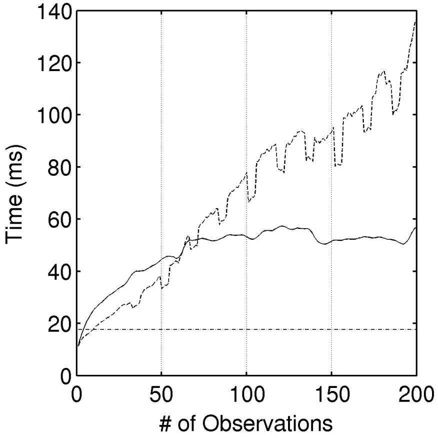


Figure: The cost-accuracy trade-off for approximating the LSQ objective function by a sum of Gaussian distributions with standard deviation of σ.

Approximation by Continuous Objective



Exploiting Parallelism

$$
\begin{aligned}
J(\mathbf{p}) &= \sum_{i=1}^{n_o} \frac{||\tilde{\mathbf{y}}(t_i) - \mathbf{y}(t_i, \mathbf{p})||^2}{2} \\
&= \sum_{i=1}^{\frac{n_o}{2}} \frac{||\tilde{\mathbf{y}}(t_i) - \mathbf{y}(t_i, \mathbf{p})||^2}{2} + \sum_{i=\frac{n_o}{2}+1}^{n_o} \frac{||\tilde{\mathbf{y}}(t_i) - \mathbf{y}(t_i, \mathbf{p})||^2}{2} \\
&= J_1(\mathbf{p}) + J_2(\mathbf{p})
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial J}{\partial \mathbf{p}}(\mathbf{p}) &= \sum_{i=1}^{n_o} \frac{||\tilde{\mathbf{y}}(t_i) - \mathbf{y}(t_i, \mathbf{p})||^2}{2} \\
&= \sum_{i=1}^{\frac{n_o}{2}} \frac{||\tilde{\mathbf{y}}(t_i) - \mathbf{y}(t_i, \mathbf{p})||^2}{2} + \sum_{i=\frac{n_o}{2}+1}^{n_o} \frac{||\tilde{\mathbf{y}}(t_i) - \mathbf{y}(t_i, \mathbf{p})||^2}{2} \\
&= \frac{\partial J_1}{\partial \mathbf{p}}(\mathbf{p}) + \frac{\partial J_2}{\partial \mathbf{p}}(\mathbf{p})
\end{aligned}
$$

CheckPointing (Petzold [2])

- Concerned with reducing memory requirements of Adjoint approach
- During forward simulation of the system, store information periodically to allow for the forward simulation to be restarted during the backward simulation, as the solution of the forward system is required.
- Alternatively, one could simulate the system in reverse, but it may be unstable

Summary

- Cost of Adjoint Method scales with the number of observations for LSQ objective functions.
- This cost is due to the observations limiting the step size taken by the solver
- In the case of DDEs, discontinuities in $\boldsymbol{\lambda}(t)$ make this cost prohibitively expensive.
- Demonstrated use of a lower order method for approximating the adjoint equations and sensitivities for ODEs.
- Demonstrated a continuous approximation to the objective function to reduce the dependence on the number of observations for DDEs.

Select References I

Y. Cao, S. Li, L. Petzold, and R. Serban.
Adjoint sensitivity analysis for differential algebraic equations: The adjoint dae system and its numerical solution.
SIAM J. Sci. Comput., 3(23):1076–1089, 2003.

Yang Cao, Shengtai Li, and Linda Petzold.
Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software.
Journal of computational and applied mathematics, 149(1):171–191, 2002.

H. Zivari-Piran.
Efficient Simulation, Accurate Sensitivity Analysis and Reliable Parameter Estimation for Delay Differential Equations.
PhD thesis, Univeristy of Toronto, 2009.

Select References II

H. Zivari-Piran and W.H. Enright.
Accurate first-order sensitivity analysis for delay differential equations: Part ii: The adjoint approach.
preprint, Department of Computer Science, University of Toronto, 2009.

Approximation by Continuous Objective

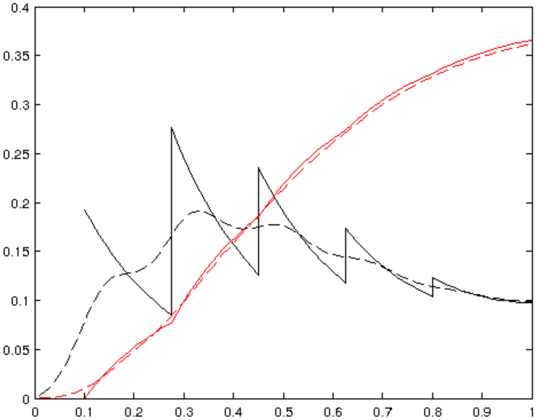


Figure: The adjoint trajectories computed using a continuous approximation with $\sigma = 0.05$ (dotted lines) and the exact adjoint trajectories (solid lines).